

AP 27W



REALNET.115A

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appellant : Jeremy Chaney
Appl. No. : 09/577,257
Filed : May 22, 2000
For : SYSTEM AND METHOD OF
PROVIDING FOR THE
CONTROL OF A MUSIC
PLAYER TO A DEVICE DRIVER
Examiner : Dennis Bonshock
Group Art Unit : 2173

CERTIFICATE OF MAILING

I hereby certify that this correspondence and all marked attachments are being deposited with the United States Postal Service as first-class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on

November 30, 2005



Eric M. Nelson, Reg. No. 43,829

APPELLANT'S REPLY BRIEF

Board of Patent Appeals and Interferences
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Appellant has the following remarks with respect to the Examiner's Answer dated October 6, 2005.

Issue

A patent application claim cannot be rejected as being obvious, unless the cited references teach or suggest each element of the claim. The issue before the board is whether Appellant's application should be allowed when the cited references (Katz and Hanson) alone or in combination do not suggest the claimed invention, e.g., a device driver interface that allows a device driver to indicate changes to a music player's graphical user interface.

Claims 1, 6, 11, 35, and 46

Appellant respectfully submits that the Examiner has failed to establish that the cited references teach or suggest at least one limitation from each of the above-listed claims. For

Appl. No. : **09/577,257**
Filed : **May 22, 2000**

example, independent Claim 1 recites a “music player” that provides “a device driver interface.” Also as claimed, the device driver provides “via the device driver interface a control object for managing music items.” Independent Claim 6 recites: “wherein the music player provides an application programming interface that enables device drivers to modify the music player’s graphical user interface” and wherein the “device drives” provides a “control object” “via the application programming interface.” Independent Claim 11 recites: “wherein the music player provides a device driver interface that enables device drivers to control the graphical user interfaces of the music player.” Independent Claim 35 recites: “providing the music player a device driver interface that enables device drivers to modify the graphical user interfaces of the music play” and “wherein the control object is provide via the device driver interface by a device driver related to a music renderer.” Independent Claim 46 includes similar types of recitations.

It appears that the Examiner has improperly construed the scope of the claim so as to eviscerate certain of the above-limitations from the claims. In the Examiner’s Answer, the Examiner stated:

Since the interpretation of the limitation is the basis for the arguments, the Examiner’s interpretation is now given. The claim, as interpreted by the examiner, pertains to a graphical user interface (GUI) interfaces with a device driver interface (not necessarily displayed) that incites a change in the GUI.
See Examiner’s Answer, p. 9.

Appellant submits that the Examiner’s characterization is incorrect and over-broad. In each of the above-referenced claims, a device driver interface is provided so that a device driver can communicate with a music player and modify a graphical user interface *of the music player*. For example, the device driver can via the device driver interface invoke predefined routines in the music player, e.g., identify window pane routine (304) and customize interface name (312) (Figure 3). With the above-definition, the Examiner has ignored the claim recitation that the device driver communicates with an application via a device driver interface. Furthermore, the Examiner has ignored that a “control object for managing music items” is provided from the device driver to the music player via the device driver interface.

Appellant fully recognizes that Hanson describes the usage of device drivers that can display information and graphical user interfaces of its own. Figures 3-8 illustrate exemplary “interactive menus” of Hanson. However, these are menus of the *device driver* and are not

Appl. No. : **09/577,257**
Filed : **May 22, 2000**

menus of the “application”, *e.g.*, the music player, as claimed. Hanson states “the peripheral device driver includes a graphical user interface (GUI) for entering of user initiated controlling commands.” *See* col. 2, lines 45-47. Figure 2 of Hanson illustrates that the GUI objects (52) are part of the device driver 44. Thus, the GUI objects (52) of Hanson are displayed by the device driver (42) and not the application software (32). There is no teaching or suggestion that the device drivers (42) transmit the GUI objects (52) to the application software (32) such as via a device driver interface as is claimed.

Appellant notes that the Examiner broadly stated that the above-limitation was described by Hanson. However, Appellant respectfully submits that the Examiner’s specific findings in advancing this statement do not truly support the Examiner’s position. In the Examiner’s Answer, the Examiner stated:

Hanson, though, discloses a dynamic device driver for a peripheral device that is capable of delivering control objects to an application. In column 2: lines 11-19, Hanson discloses a peripheral device that is connected to a host computer. In column 2: lines 40-44, Hanson explains that the peripheral could be an audio component. In column 2: lines 45-50, Hanson further explains that the peripheral’s device driver includes a graphical interface for handling user-initiated controlling commands and for displaying the status of the peripheral device as well as a list of predefined user-selectable options related to the peripheral device. *See* Examiner’s Answer, p. 10.

With respect to each of the above-statements, Appellant agrees. However, nowhere in the cited sections is there a teaching or suggestion that the device driver (42) of Hanson communicates with the application software (32) of Hanson to as to indicate a change to the display of an interface of the application software (32). In Hanson, the device driver (42) is merely displaying on its own graphical user interfaces. It is not transmitting the GUI objects (52) to the application software (32).

In the Examiner’s Answer, the Examiner also stated:

Hanson still further explains in column 5: lines 13-22 that the graphical interface objects provided by the peripheral driver software can be incorporated into the menus of the application software running on the host computer. *See* Examiner’s Answer, page 8.

For convenience of discussion, the passage cited by the Examiner is reproduced below.

Another embodiment, not shown, incorporates the GUI objects 52 into the menus of the application software 32 running on the operating system. For example, a

Appl. No. : 09/577,257
Filed : May 22, 2000

printer options window of the common command windows for Word 6.0 of a printer connected to the host computer system through a standard device driver and a printer connected to the host through the dynamic device driver 42 would be visually the same and may include similar printer options.

Appellant respectfully submits that this passage suggests that an application can be built, such as at time of compilation, to physically include the objects that may then be used to control printer functions. The foregoing passage does not teach or suggest providing a well-defined device driver interface in an application, wherein the interface allows device drivers to provide graphical control objects to the application. In the embodiment described above, the menus that are controlled by the Hanson system could be redesigned such that the menus could be put under control of an application instead of the device driver. Hanson fails to teach or suggest that control objects can be transmitted via a device driver interface from a device driver to an application. Appellant notes that this was fully discussed in the Appeal Brief and the Examiner entirely failed to comment on this issue in the Examiner's Answer.

In the Examiner's Answer, the Examiner also stated:

The examiner respectfully contends that Hanson does teach this device driver interface, in column 4, lines 20-57, where he teaches a dynamic device driver containing both an OS specific device driver portion and an OS independent device driver portion, the OS independent device driver portion interfacing with the GUI either locally at the host computer system or through a local area network. The OS independent device driver portion includes information regarding peripheral device operation, peripheral specific data objects, and GUI objects. The GUI objects being supplied as a means for the user to interface with the device drover and in turn the device.

See Examiner's Answer, p. 11.

Appellant acknowledges that the device driver (42) of Hanson displays graphical user interfaces of its own. Furthermore, Appellant acknowledges that Hanson states that device driver (42) includes an OS specific device driver portion (33) that acts as a communication layer between the operating system and the OS independent device driver portion (34). *See* col. 4, lines 36-39. It is also noted that the OS independent device driver portion (34) can reside either in the host computer system (10), in the peripheral device or in a device connected to a network. *See* col. 4, lines 39-45. However, the cited sections fail to suggest an interface for transmitting control objects from the device driver (42) to the application software (32) so as to indicate a

Appl. No. : 09/577,257
Filed : May 22, 2000

change of the display of the application's graphical interfaces. There simply is no suggestion of this in the cited sections.

Moreover, in the Examiner Answer, the Examiner stated:

From page 9 of the Appeal Brief, from the fifth paragraph, the Appellant argues that "Hanson fails to teach or suggest an application provide a device driver interface for controlling its graphical user interface."

The examiner respectfully contends that Hanson does teach, in column 4, line 45 through column 5, line 22, the GUI objects supplied OS independent device driver being displayed in menus of the application software running on the operating system. These menus being for use in operating the peripheral device. *See* Examiner's Answer, p. 12.

Appellant respectfully submits that the cited section, i.e., col. 4, line 45-col. 5, line 22 of Hanson, never states or suggests that the GUI objects are used to indicate a change to the GUI of the application software (32). The GUI objects are used to provide and display interfaces of *the device driver*. Furthermore, a device driver that displays menus, as in Hanson, does not anticipate or make obvious the claimed invention. In one example of using the claimed invention, it is possible to modify selected aspects of a GUI of an application. For example, referring to Figure 6 it is possible for a device driver to change the name of a "begin transfer" button of the music player to a "begin CD burn" button. *See* Application, p. 9, lines 21-29. In Hanson, the device driver (42) is responsible for displaying its own graphical user interfaces—there is no suggestion that it could indicate a change to a graphical user interface of the application software (32).

Thus, Appellant respectfully submits that the Examiner has wholly failed to establish that the prior art suggests the desirability of having an application provide a device driver interface to allow device drivers to indicate changes to the graphical user interfaces of the application as opposed to the device driver only indicating changes to the device driver itself.

Claims 16 and 17

One embodiment of the invention allows a device driver to "rename" control objects that are operative to be used by a user to control the operation of a music renderer, e.g., MP3 player. For example, referring to Figure 6 it is possible for a device driver to change the name of a "begin transfer" button to a "begin CD burn" button. *See* Application, p. 9, lines 21-29.

Appl. No. : **09/577,257**
Filed : **May 22, 2000**

In the Examiner's Answer, the Examiner stated:

The Examiner respectfully contends that menus can display a specific set of GUI objects for each specific peripheral device. Should one device be left on the display a new device be selected for use the OS independent portion will supply an instruction to rename the items in the menu to concur with the currently selected peripheral device (see column 5, lines 12-43).

See Examiner's Answer, p. 15.

Appellant has found it difficult to discern the Examiner's point with respect to the above. It appears the Examiner is generally stating that the data elements that are displayed by the Hanson device driver will change based upon the user's actions. This construction of the Examiner's position appears to be supported by his later remarks which state:

From page 12 of the Appeal Brief, from the third paragraph, the Appellant argues that "The Examiner has failed to particularly identify how control objects of Hanson would initially be named by the application.

The examiner respectfully contends that the prior art selection of a printer to use from the "available printers" display would give the initial selection, providing its own distinct displayed menus (see column 5, lines 12-43).

See Examiner's Answer, p. 15.

Appellant respectfully submits that the responsibility for displaying and listing available printers is that of the device driver. Appellant respectfully submits that the cited section fails to even tangentially suggest that the device driver (42) can rename a control object of the application software (32). Appellant respectfully submits that the Examiner's analysis has failed to explain: 1) how a list of available printers is a control object of the application software (32) as opposed to that of device driver (44), 2) the device driver (44) "renamed" the list the available printers, and 3) how the list of available printers is adapted to be used by a user to control the operation of a music renderer that is configured to play the music items, as is claimed.

Claim 39

With respect to independent Claim 39, one embodiment of Appellant's invention is generally directed to a device driver interface that may be used to provide control of a graphical interface with a device driver of a portable music player. Claim 39 recites: "executing a music player that is executing in a computer and that displays a graphical interface comprising information about music items; displaying, in response to a user request for transferring a music

Appl. No. : 09/577,257
Filed : May 22, 2000

item from the computer to a portable music player device, a graphical interface for managing the content of the portable music player device; and assigning an object in the graphical interface with a device driver of the portable music player device.” Thus, as is recited, a device driver can be assigned control of an object that is in the graphical interface of a music player.

Appellant respectfully submits that the cited prior art fails to teach or suggest “assigning an object in the graphical interface with a device driver of the portable music player device.” Katz is generally directed to a software application for displaying music. The cited references wholly fail to teach or suggest allowing a device driver to be assigned control of an object in a graphical interface of a music player. Hanson describes that a printer device driver may have certain graphical user interfaces of its own. For example, Figures 3-8 of Hanson describe various graphical user interfaces that are displayed by the device driver (42)—not the application software (32). Appellant notes that in the Examiner’s Answer, the Examiner wholly failed to identify where in Hanson the device driver (44) assigns an object in the graphical interface of the application. In Hanson, the GUI objects 52 relate to the interfaces of the device driver (44) and not the application software (32). Hanson fails to teach or suggest that the *device driver* (44) assign control of an object in the graphical interface of the *application software* (32), such as the claimed “music player.”


Summary

Since the cited prior art fails to teach or suggest in isolation or in combination at least one limitation from each of the above-listed claims, Appellant respectfully submits that the above-claims are in condition for immediate allowance.

Respectfully submitted,

KNOBBE, MARTENS, OLSON & BEAR, LLP

Dated: 11/30/2005

By: 
Eric M. Nelson
Registration No. 43,829
Attorney of Record
Customer No. 20,995
(619) 235-8550